AMENDMENTS TO THE CLAIMS

This listing of claim will replace all prior versions and listings of claim in the application.

1. (currently amended) A method for detecting whether a routine has stalled, comprising ~~the steps of~~:

accessing existing code for a first routine;

automatically modifying said existing code ~~to include~~ by changing said existing code and adding new code; and

using said new code and said changed existing code to determine if said first routine has stalled.

2. (currently amended) A method according to claim 1, wherein:

said existing code is modified prior to starting execution of said existing code;

said existing code is object code; and

said new code is object code.

3. (currently amended) A method according to claim 2, further comprising ~~the step of~~:

receiving a user created rule, said rule identifies said first routine and an interval, said step of automatically modifying is performed in response to said rule, said first routine is considered to be stalled after a determination that said first routine has been running for at least as long as said interval.

4. (currently amended) A method according to claim 1, wherein said step of automatically modifying comprises ~~the steps of~~:

adding object code for a timing mechanism to said existing code;

~~adding code for starting said timing mechanism to said existing object code;~~

~~adding code for stopping said timing mechanism to said existing object code;~~

adding a first instruction to said existing code for said first routine, said first instruction causes ~~the execution of said code for~~ a starting of said timing mechanism for said first routine, said first instruction is object code; and

adding a second instruction to said existing object code for said first routine, said second instruction causes a ~~the execution of said code for~~ stopping of said timing mechanism for said first routine, said second instruction is object code.

5. (currently amended) A method according to claim 4, wherein:
said second instruction is added such that it is executed at all exits of said first routine.

6. (currently amended) A method according to claim 4, wherein:
said first routine is a method associated with an object[[.]]; and
said changing said existing code includes adding code to said method to call said new code.

7. (original) A method according to claim 4, wherein:
said first routine is a thread.

8. (original) A method according to claim 4, wherein:
said first routine is one of a plurality of routines that comprise a process.

9. (currently amended) A method according to claim 1, wherein said step of using comprises ~~the steps of~~:

receiving an indication that said first routine has started;

starting a timing mechanism in response to said step of receiving;

receiving an indication that said first routine has completed[[,]] if said first routine has completed, said indication is caused by said changed existing code;

stopping said timing mechanism in response to receiving said indication that said first routine has completed, if said first routine has completed; and

reporting said first routine as stalled if said timing mechanism is not stopped prior to ~~a determination that~~ said timing mechanism ~~is~~ being overdue for said first routine.

10. (currently amended) A method for detecting whether a routine has stalled, comprising: ~~A method according to claim 9, wherein said step of using further comprises the steps of:~~

accessing existing code for a first routine;

automatically modifying said existing code to include new code; and

using said new code to determine if said first routine has stalled, said step of using comprises:

receiving an indication that said first routine has started,

starting a timing mechanism in response to said step of receiving,

receiving an indication that said first routine has completed if said first routine has completed,

stopping said timing mechanism in response to receiving said indication that said first routine has completed,

accessing a current time[[;]],

verifying that said first routine is not known to be stalled or completed[[;]],

accessing ~~said~~ a due time[[;]], ~~and~~

determining whether ~~first~~ due time is earlier than said current time, said timing mechanism is overdue if ~~said step of determining concludes that~~ said ~~first~~ due time is earlier than said current time[[.]], and

reporting said first routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

11. (original) A method according to claim 9, wherein said step of stopping said timing mechanism comprises the steps of:

determining whether said first routine has been reported as being stalled;

changing said reporting to no longer indicate that said first routine is stalled if said step of determining concludes that said first routine has been reported as being stalled; and

stopping said timing mechanism if said first routine has not been reported as being stalled.

12. (original) A method according to claim 9, wherein:

said first routine is a thread executing a method, said indication that said first routine has completed indicates that said thread has exited said method.

13. (currently amended) A method for detecting whether a routine has stalled, comprising ~~the steps of~~:

receiving an indication that a particular routine has started ~~is running,~~ said particular routine is one of a plurality of routines that comprise a process; ~~and~~

~~automatically determining whether said particular routine has stalled.~~

receiving an indication that said particular routine has completed, if said particular routine has completed; and

reporting said particular routine as stalled if said indication that said particular routine has completed is not provided prior to being overdue.

14. (currently amended) A method according to claim 13, ~~wherein~~ further comprising:

~~said particular routine is a thread.~~

starting a timing mechanism for said particular routine in response to said indication that said particular routine has started;

calculating a due time based on a current time and a time interval; and

determining whether said indication that said first routine has completed is provided prior to said due time, said indication that said first routine has completed is overdue if provided after said due time.

15. (currently amended) A method according to claim 13, wherein:

said particular routine is a thread;

said plurality of routines are threads; and

at least two or more of said threads, including said particular routine, are run concurrently.

16. (original) A method according to claim 13, wherein:

said particular routine is a method

17. (currently amended) A method according to claim 16, wherein: ~~further comprising the step of:~~

~~receiving an indication of said method, said step of determining whether said particular routine is stalled includes detecting~~ ~~a situation when~~ said steps of receiving an indication that a particular routine has started and receiving an indication that a particular routine has completed include detecting whether a thread enters said method and does not return within an approximation of an expected time frame.

18. (cancelled)

19. (currently amended) A method according to claim 13 [[18]], wherein:

said indication that said particular routine has started is received from within said particular routine; and

said indication that said particular routine has completed is received from within said particular routine.

20. (currently amended) A method according to claim 14 [[18]], further comprising: ~~the step of:~~

automatically changing existing code for said particular routine; and

automatically adding new code to said existing code for said particular routine, said new code performs said step of starting a timing mechanism~~, stopping said timing mechanism and reporting~~.

21. (currently amended) A method according to claim 14 [[18]], further comprising the step of:

automatically modifying existing object code for said particular routine ~~in order to add~~ to call new object ~~code to said existing object code for said particular routine~~, said new object code performs said step of starting a timing mechanism~~, stopping said timing mechanism and reporting~~.

22. (currently amended) A method according to claim 14 [[18]], wherein said determining whether said indication that said first routine has completed is provided prior to said due time comprises: ~~step of determining whether said particular routine has started further comprises the steps of:~~

accessing a new current time;

verifying that said particular routine is not known to be stalled or completed;

accessing [[a]] said due time; and

determining whether said due time is earlier than said new current time~~, said timing mechanism is overdue if it is determined that said due time is earlier than said current time~~.

23. (cancelled)

24. (currently amended) A method according to claim 14 [[18]], wherein said step of calculating a due time comprises: ~~starting said timing mechanism comprises the steps of:~~

receiving a threshold;

accessing [[a]] said current time;

determining ~~a first~~ said due time based on said threshold and said current time; and

adding a indication of said particular routine and said ~~first~~ due time to a set of due times for other routines~~, said timing mechanism is overdue after said timing mechanism determines that said due time has been exceeded~~.

25. (currently amended) A method according to claim 24, wherein said step determining comprises: ~~of stopping said timing mechanism comprises the steps of:~~

determining whether said particular routine has been reported as being stalled;

changing said reporting to no longer indicate that said particular routine is stalled if said particular routine has been reported as being stalled; and

removing said indication of said particular routine and said due time from said set of items if said particular routine has not been reported as being stalled.

26. (original) A method according to claim 25, further comprising the step of: repeatedly evaluating said set of due times to determine if any of said due times have passed.

27. - 29. (cancelled)

30. (currently amended) A method according to claim 13 [[29]], wherein:

said particular routine is an instance of a defined routine; and

said step of reporting includes incrementing a counter that represents a number of instances of said defined routine that are currently stalled and reporting said number of instances of said defined routine that are currently stalled.

31. (currently amended) A method according to claim 13 [[29]], wherein:

said particular routine is an instance of a defined routine; and

said step of reporting includes determining and reporting how many instances of said defined routine were stalled at a specified time.

32. (currently amended) A method according to claim 13 [[29]], wherein:

said particular routine is an instance of a defined routine; and

said step of reporting includes receiving a customizable specified time period and reporting how many instances of said defined routine were stalled during said specified time period.

33. - 38. (cancelled)

39. (currently amended) A method for detecting whether a method has stalled, comprising: ~~the steps of:~~

receiving an indication that a particular method of an object is running; and

automatically determining whether said method ~~thread~~ has stalled by detecting whether a thread entered said method and did not return within an approximation of an expected time frame.

40. (cancelled)

41. (currently amended) A method according to claim 39, further comprising the step of:

automatically ~~modifying~~ changing existing object code for said ~~particular~~ method ~~in order to add~~ and adding new object code, said new object code performs at least a portion of said step of automatically determining.

42. (cancelled)

43. (currently amended) A method according to claim 39, further comprising: ~~42, wherein said step of stopping said timing mechanism comprises the steps of:~~

determining whether said particular method has been previously reported as being stalled; and

changing said reporting to no longer indicate that said particular method is stalled if it is determined that said particular method has been reported as being stalled and said particular method is no longer stalled.~~; and~~

~~stopping said timing mechanism if said particular method has not been reported as being stalled.~~

44. (cancelled)

45. (original) A method for detecting whether a routine has stalled, comprising the steps of:

receiving an indication that a first routine has started;

starting a timing mechanism in response to said indication that said first routine has started;

receiving an indication that said first routine has completed, if said first routine has completed;

stopping said timing mechanism in response to receiving said indication that said first routine has completed; and

reporting said first routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

46. - 47. (cancelled)

48. (original) A method according to claim 45, further comprising the steps of:

accessing a current time;

verifying that said first routine is not known to be stalled or completed;

accessing a due time for said first routine; and

determining whether said due time is earlier than said current time, said timing mechanism is overdue if said step of determining concludes that said first due time is earlier than said current time.

49. (original) A method according to claim 45, wherein said step of stopping said timing mechanism comprises the steps of:

determining whether said first routine has been reported as being stalled;

changing said reporting to no longer indicate that said first routine is stalled if said first routine has been reported as being stalled; and

stopping said timing mechanism if said first routine has not been reported as being stalled.

50. (original) A method according to claim 45, wherein:

said first routine is an instance of a defined routine; and

said step of reporting includes incrementing a counter that represents a number of instances of said defined routine that are stalled and reporting said number of instances of said defined routine that are stalled.

51. (original) A method according to claim 45, wherein:

said first routine is an instance of a defined routine;

said step of reporting includes receiving a customizable specified time period and reporting how many instances of said defined were stalled during said specified time period.

52. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors which when executing the code perform ~~to perform~~ a method comprising the steps of:

accessing ~~existing~~ object code for a first ~~routine~~ method;

automatically changing said ~~existing~~ object code for said first method to call new code; ~~modifying said existing code to include new code;~~ and

using said new code to determine if said first ~~routine~~ method has stalled.

53. - 55. (cancelled)

56. (currently amended) One or more processor readable storage devices according to claim 52, wherein said step of using comprises ~~the steps of~~:

receiving an indication that said first ~~routine~~ method has started;

starting a timing mechanism in response to said step of receiving;

receiving an indication that said first ~~routine~~ method has completed, if said first method has completed;

stopping said timing mechanism in response to receiving said indication that said first method has completed; and

reporting said first method as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue for said first method.

57. (cancelled)

58. (currently amended) An apparatus, comprising:

one or more storage devices; and

one or more processors in communication with said one or more storage devices, said one or more processors perform a method comprising: ~~the steps of:~~

accessing ~~existing~~ object code for a first routine,

automatically ~~modifying said existing code to include~~ changing said object code and adding new code, and

using said changed object code and said new code to determine if said first routine has stalled.

59. - 61. (cancelled)

62. (original) An apparatus according to claim 58, wherein said step of using comprises the steps of:

receiving an indication that said first routine has started;

starting a timing mechanism in response to said step of receiving;

receiving an indication that said first routine has completed, if said first routine has completed;

stopping said timing mechanism in response to receiving said indication that said first routine has completed; and

reporting said first routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

63. (cancelled)

64. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors which when executing the code perform ~~to perform~~ a method comprising the steps of:

receiving an indication that a particular routine has started; ~~is running, said particular routine is one of a plurality of routines that comprise a process; and~~

~~automatically determining whether said particular routine has stalled.~~

receiving an indication that said particular routine has completed, if said particular routine has completed; and

reporting said particular routine as stalled if said indication that said particular routine has completed is not provided prior to being overdue.

65. - 66. (cancelled)

67. (currently amended) One or more processor readable storage devices according to claim 64, wherein: ~~66,further comprising~~ ~~the step of:~~

~~receiving an indication of said method, said step of determining whether said particular routine is stalled includes detecting~~ ~~a situation when~~ said steps of receiving an indication that a particular routine has started and receiving an indication that a particular routine has completed include detecting whether a thread enters ~~said~~ a method and does not return within an approximation of an expected time frame.

68. (currently amended) One or more processor readable storage devices according to claim 64, ~~wherein:~~ further comprising:

starting a timing mechanism for said particular routine in response to said indication that said particular routine has started;

calculating a due time based on a current time and a time interval; and

determining if said indication that said particular routine has completed is provided prior to said due time, said indication that said particular routine has completed is overdue if provided after said due time.

~~said indication that a particular routine is running is an indication that said particular routine has started; and~~

~~said step of determining whether said particular routine has stalled comprises the steps of:~~

~~starting a timing mechanism in response to said step of receiving an indication that a particular routine is running,~~

~~receiving an indication that said particular routine has completed, if said particular~~

~~routine has completed,~~

~~stopping said timing mechanism in response to receiving said indication that said particular routine has completed, and~~

~~reporting said particular routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.~~

69. (currently amended) One or more processor readable storage devices according to claim 68, wherein said method further comprises ~~the step of~~:

automatically ~~modifying~~ changing existing object code for said particular routine in order to ~~add~~ point to new object code ~~to said existing object code for said particular routine~~, said new object code performs said step of starting a timing mechanism~~, stopping said timing mechanism and reporting~~.

70. (currently amended) One or more processor readable storage devices according to claim 68, ~~wherein said step of stopping said timing mechanism comprises the steps of:~~ further comprising:

determining whether said particular routine has been reported as being stalled;

changing said reporting to no longer indicate that said particular routine is stalled if it is determined that said particular routine has been reported as being stalled; and

stopping said timing mechanism if said particular routine has not been reported as being stalled.

71. (currently amended) One or more processor readable storage devices according to claim 64, wherein:

said particular routine is an instance of a defined routine; and

said step of ~~automatically determining~~ reporting includes incrementing a counter that represents a number of instances of said defined routine that are currently stalled and reporting said number of instances of said defined routine that are currently stalled.

72. (currently amended) An apparatus, comprising:

one or more storage devices; and

one or more processors in communication with said one or more storage devices, said one or more processors perform a method comprising ~~the steps of~~:

receiving an indication that a particular routine is running, said particular routine is one of a plurality of routines that comprise a process, ~~and~~

automatically determining ~~whether~~ that said particular routine has not stalled,[[.]]

determining that said particular routine has been previously reported as being stalled, and

changing said reporting to no longer indicate that said particular routine is stalled.

73. -74. (cancelled)

75. (currently amended) An apparatus according to claim 74, wherein: ~~further comprising the step of:~~

~~receiving an indication of said method,~~ said step of automatically determining ~~whether~~ that said particular routine ~~is~~ has not stalled includes detecting ~~a situation when~~ that a thread enters ~~said~~ a method and ~~does not~~ returns within an approximation of an expected time frame.

76. (cancelled)

77. (original) An apparatus according to claim 72, wherein said method further comprises the step of:

automatically ~~modifying~~ changing existing object code for said particular routine ~~in order to add~~ and adding new object code to said existing object code ~~for said particular routine~~, said new object code performs said step of automatically determining ~~starting a timing mechanism, stopping said timing mechanism and reporting~~.

78. -79. (cancelled)